

A global convergent quasi-Newton method for systems of monotone equations

Zixin Chen · Wanyou Cheng · Xiaoliang Li

Received: 13 March 2013 / Published online: 16 July 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract In this paper, we propose a quasi-Newton method for solving systems of monotone equations. The method is a combination of the Broyden method and the hyperplane projection method. Under appropriate conditions, we prove that the proposed method is globally convergent. Preliminary numerical results show that the proposed method is promising.

Keywords Monotone equations · Hyperplane projection method · Global convergence

Mathematics Subject Classification 90C25 · 90C33

1 Introduction

In this paper, we consider the problem of finding a solution of the nonlinear system of equations

$$F(x) = 0, \quad (1.1)$$

The work was supported by the NSF of China via grant (11071087, 11101081).

Z. Chen

College of City, Dongguan University of Technology, Dongguan 523808, China
e-mail: zxchen@tom.com

W. Cheng (✉) · X. Li

College of Computer, Dongguan University of Technology, Dongguan 523808, China
e-mail: chengwanyou421@yahoo.com.cn

X. Li

e-mail: xli@dgut.edu.cn

where $F : R^n \rightarrow R^n$ is continuous and monotone. By monotonicity, we mean

$$\langle F(x) - F(y), x - y \rangle \geq 0, \quad \forall x, y \in R^n.$$

Nonlinear monotone equations have many practical background such as the first order necessary condition of the unconstrained convex optimization problem and the sub-problems in the generalized proximal algorithms with Bregman distances [9]. Some monotone variational inequality problems can also be converted into the form of the problem by means of fixed point maps or normal maps if the underlying function satisfies some coercive conditions [12].

Many methods for solving (1.1) fall into the class of the quasi-Newton methods since they converge rapidly from sufficiently good initial guess. Since Broyden [1] proposed the first quasi-Newton method for solving nonlinear equations, there has been significant progress in the theoretical study on quasi-Newton methods, especially in local convergence analysis [2, 3]. To ensure global convergence, some line search strategy for some merit function are used. Recently, Solodov and Svaiter [10] presented a Newton-type algorithm for solving systems of monotone equations. By using hybrid projection method, they showed that their method converges globally. For nonlinear equations, Griewank [4] obtained a global convergence results for Broyden's rank one method. By introduce a new line search process, Li and Fukushima [7] have developed a globally convergent Broyden-like method for solving nonlinear equations and [6] presented a globally convergent Gauss-Newton-based BFGS method for solving symmetric nonlinear equations. The method in [6, 7] is not norm descent. Gu, Li, Qi and Zhou [5] generalized the method in [6] and proposed a globally convergent and norm descent BFGS method for solving symmetric equations. Quite recently, Zhou and Li [13] proposed a global convergence BFGS method for systems of monotone equations without use of merit functions. We refer to papers [8, 11] for a review on recent advances in this area.

In this paper, based on the hyperplane projection method [10], we propose a quasi-Newton method for solving systems of monotone equations without use of merit functions. The method is a combination of the Broyden method and the hyperplane projection method [10]. Under appropriate conditions, we show that the proposed method is globally convergent. Preliminary numerical results show that the method is promising.

The paper is organized as follows. In Sect. 2, after simply recalling hyperplane projection method, we present the algorithm. In Sect. 3, we establish the global convergence of the algorithm. We report some numerical results in the last section.

2 Algorithm

In this section, we describe the method in detail. Firstly, let us first recall the hyperplane projection method in [10]. Note that by the monotonicity of F , for any \bar{x} such that $F(\bar{x}) = 0$, we have

$$\langle F(z_k), \bar{x} - z_k \rangle \leq 0.$$

Let x_k be the current iterate. By performing some kind of line search procedure along a direction \vec{d}_k , a point $z_k = x_k + \alpha_k \vec{d}_k$ can be computed such that

$$\langle F(z_k), x_k - z_k \rangle > 0.$$

Thus the hyperplane

$$\mathcal{H}_k = \{x \in R^n \mid \langle F(z_k), x - z_k \rangle = 0\}$$

strictly separates the current iterate x_k from zeros of (1.1). Therefore, it is reasonable to let the next iterate x_{k+1} be the projection of x_k onto the hyperplane.

Now, we state the steps of the algorithm as follow.

Algorithm 2.1 (Broyden method)

Step 1. Given an initial point $x_0 \in R^n$ and constants $\beta \in (0, 1)$, $\eta \in (0, 1)$, $\xi \in (0, 1)$ and $0 < \sigma_{\min} < \sigma_{\max}$. Given the initial steplength $\sigma_0 = 1$, $B_0 = I$ (the identity matrix) and $d_0 = -F(x_0)$. Let $k := 0$.

Step 2. Stop if $\|F(x_k)\| = 0$.

Step 3. Determine steplength $\alpha_k = \sigma_k \beta^{m_k}$ such that m_k is the smallest nonnegative integer m satisfying

$$-\langle F(x_k + \sigma_k \beta^m d_k), d_k \rangle \geq \eta \sigma_k \beta^m \|F(x_k + \sigma_k \beta^m d_k)\| \|d_k\|^2. \quad (2.1)$$

Let $z_k = x_k + \alpha_k d_k$. Stop if $\|F(z_k)\| = 0$.

Step 4. Compute the projection of x_k on \mathcal{H}_k by

$$x_{k+1} = x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k). \quad (2.2)$$

Stop if $\|F(x_{k+1})\| = 0$.

Step 5. Compute B_{k+1} by the following Broyden update formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{\|s_k\|^2} \quad (2.3)$$

where $s_k = x_k - x_{k-1}$ and $y_k = F(x_k) - F(x_{k-1})$.

Step 6. Compute d_{k+1} by solving the linear equation

$$B_{k+1} d_{k+1} = -F(x_{k+1}). \quad (2.4)$$

If the system (2.4) is not solvable or the condition

$$d_{k+1}^T F(x_{k+1}) < -\xi \|F(x_{k+1})\|^2$$

is not satisfied, set $d_{k+1} = -F(x_{k+1})$ and $\sigma_{k+1} = \max\{\sigma_{\min}, \min\{\frac{\|s_k\|^2}{s_k^T y_k}, \sigma_{\max}\}\}$; else set $\sigma_{k+1} = 1$.

Step 7. Let $k := k + 1$. Go to Step 3.

Remark It is easy to see from Step 6 of Algorithm 2.1 that

$$-F_k^T d_k \geq \xi \|F_k\|^2. \quad (2.5)$$

Therefore after a finite number of reductions of α_k , the line search condition (2.1) necessarily holds. Consequently, Algorithm 2.1 is well-defined.

3 Convergence property

This section is devoted to the global convergence of Algorithm 2.1. To establish global convergence of Algorithm 2.1, we need the following assumption.

Assumption 3.1 (1) ∇F is Lipschitz continuous on R^n , i.e., there is a constant $L > 0$ such that

$$\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|, \quad \forall x, y \in R^n,$$

where ∇F denotes the Jacobian of F .

(2) $\nabla F(x)$ is nonsingular for every $x \in R^n$.

Before proving global convergence of Algorithm 2.1, we first give three preliminary lemmas. The following lemma is from [10].

Lemma 3.1 Let F be monotone and $x, y \in R^n$ satisfy $\langle F(y), x - y \rangle > 0$. Let

$$x^+ = x - \frac{\langle F(y), x - y \rangle}{\|F(y)\|^2} F(y).$$

Then for any $\bar{x} \in R^n$ such that $F(\bar{x}) = 0$, it holds that

$$\|x^+ - \bar{x}\|^2 \leq \|x - \bar{x}\|^2 - \|x^+ - x\|^2.$$

Define

$$\delta_k = \frac{\|y_k - B_k s_k\|}{\|s_k\|} \quad \text{and} \quad A_{k+1} = \int_0^1 \nabla F(x_k + \tau s_k) d\tau.$$

Then by the mean-value theorem, we have $y_k = A_{k+1} s_k$ and hence

$$\delta_k = \frac{\|(A_{k+1} - B_k) s_k\|}{\|s_k\|}.$$

Moreover, by the update formula (2.3), we have

$$B_{k+1} = B_k + \frac{(A_{k+1} - B_k) s_k s_k^T}{\|s_k\|^2}.$$

In a similar way to Lemma 2.6 in [7], it is not difficult to prove the following useful lemma.

Lemma 3.2 Suppose that Assumption 3.1 holds and the sequence $\{x_k\}$ generated by Algorithm 2.1 is bounded. If

$$\sum_{k=0}^{\infty} \|s_k\|^2 < \infty, \quad (3.1)$$

then there is a subsequence of $\{\delta_k\}$ tending to zero.

Lemma 3.3 Suppose that Assumption 3.1 holds and the sequence $\{x_k\}$ generated by Algorithm 2.1 is bounded. If (3.1) holds, then there exist an infinite set K_1 and a constant $C_1 > 0$ such that

$$\xi \|F(x_k)\| \leq \|d_k\| \leq C_1 \|F(x_k)\| \quad (3.2)$$

for all $k \in K_1$ large enough.

Proof By Lemma 3.2, there is a subsequence $\{\delta_k\}_{k \in K}$ of $\{\delta_k\}$ converging to zero. Since $\{x_k\}_{k \in K}$ is bounded, there exists an infinite set $K_1 \subset K$ such that $\lim_{k \in K_1} x_k = \bar{x}$. By (3.1) and the definition of A_{k+1} , it is clear that $\{A_{k+1}\}_{k \in K_1}$ tends to $\nabla F(\bar{x})$. By the nonsingularity of $\nabla F(\bar{x})$, there exists a constant $M_1 > 0$ such that $\|A_{k+1}^{-1}\| \leq M_1$ for all $k \in K_1$ sufficiently large. Thus by Step 6 of Algorithm 2.1 and the definition of δ_k , we have

$$\begin{aligned} \|d_k\| &\leq \max\{\|F(x_k)\|, \|A_{k+1}^{-1}((A_{k+1} - B_k)d_k - F(x_k))\|\} \\ &\leq \max\{\|F(x_k)\|, M_1(\delta_k \|d_k\| + \|F(x_k)\|)\}. \end{aligned}$$

Since $\lim_{k \in K_1} \delta_k = 0$, the last inequality implies that there is a constant $C_1 > 0$ such that for all $k \in K_1$ sufficiently large

$$\|d_k\| \leq C_1 \|F(x_k)\|. \quad (3.3)$$

On the other hand, applying Cauchy-Schwarz inequality to (2.5), we obtain

$$\|d_k\| \geq \xi \|F_k\|.$$

The last inequality together with (3.3) implies (3.2). \square

Now we establish a global convergence theorem for Algorithm 2.1.

Theorem 3.1 Suppose that Assumption 3.1 holds. Let $\{x_k\}$ be generated by Algorithm 2.1. Suppose that F is monotone and that the solution set of (1.1) is not empty. Then for any \bar{x} such that $F(\bar{x}) = 0$, it holds that

$$\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2.$$

In particular, $\{x_k\}$ is bounded. Furthermore, it holds that either $\{x_k\}$ is finite and the last iterate is a solution, or the sequence is infinite and $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$. Moreover, $\{x_k\}$ converges to some solution of (1.1).

Proof We first note that if the algorithm terminates at some iteration k , then $\|F(z_k)\| = 0$ or $\|F(x_k)\| = 0$. This means that x_k or z_k is a solution of (1.1).

Suppose that $\|F(z_k)\| \neq 0$ and $\|F(x_k)\| \neq 0$ for all k . Then an infinite sequence $\{x_k\}$ is generated. It follows from (2.1) that

$$\langle F(z_k), x_k - z_k \rangle = -\alpha_k \langle F(z_k), d_k \rangle \geq \eta \|F(z_k)\| \alpha_k^2 \|d_k\|^2 > 0. \quad (3.4)$$

Let \bar{x} be any solution such that $F(\bar{x}) = 0$. By (2.2), (3.4) and Lemma 3.1, we obtain

$$\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2. \quad (3.5)$$

Hence the sequence $\{\|x_k - \bar{x}\|^2\}$ is decreasing and convergent. In particular, the sequence $\{\|x_k - \bar{x}\|\}$ is convergent and the sequence $\{x_k\}$ is bounded. Again by (3.5), we have

$$\|x_{k+1} - x_k\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2. \quad (3.6)$$

Summing both side of (3.6), since the sequence $\{\|x_k - \bar{x}\|^2\}$ is convergent, we have (3.1). In particular, we have

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (3.7)$$

By (2.2) and (3.4), we obtain

$$\|x_{k+1} - x_k\| = \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|} \geq \eta \alpha_k^2 \|d_k\|^2.$$

The last inequality together with (3.7) implies

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0. \quad (3.8)$$

Now we consider the following two possible cases:

- (i) $\lim_{k \rightarrow \infty} \inf \|F(x_k)\| = 0$.
- (ii) $\lim_{k \rightarrow \infty} \inf \|F(x_k)\| = \epsilon > 0$.

If (i) holds, by the continuity of F and the boundedness of $\{x_k\}$, it is clear that the sequence $\{x_k\}$ has some accumulation point \hat{x} such that $F(\hat{x}) = 0$. From (3.5), it holds that the sequence $\{\|x_k - \hat{x}\|\}$ is convergent, and since \hat{x} is an accumulation point of $\{x_k\}$, it must hold that $\{x_k\}$ converges to \hat{x} .

If (ii) holds. In this case, by the boundedness of $\{x_k\}$ and the continuity of F , there exists a positive constant C such that

$$\frac{\epsilon}{2} \leq \|F(x_k)\| \leq C. \quad (3.9)$$

On the other hand, by Assumption 3.1, the boundedness of $\{x_k\}$ and (3.1), then Lemma 3.3 holds. Lemma 3.3 and (3.2) and (3.9) implies that $\{d_k\}_{k \in K_1}$ is bounded from below. Hence, from (3.8) we get that

$$\lim_{k \in K_1, k \rightarrow \infty} \alpha_k = 0.$$

By the line search rule, we have for all $k \in K_1$ sufficiently large, $\sigma_k \beta^{m_k-1}$ will not satisfy (2.1). This means

$$-\langle F(x_k + \sigma_k \beta^{m_k-1} d_k), d_k \rangle < \eta \sigma_k \beta^{m_k-1} \|F(x_k + \sigma_k \beta^{m_k-1} d_k)\| \|d_k\|^2. \quad (3.10)$$

The boundedness of $\{x_k\}_{k \in K_1}$ implies that there exist an accumulation point \hat{x} and an infinite index set $K_2 \subset K_1$ such that $\lim_{k \in K_2} x_k = \hat{x}$. Since the sequence $\{d_k\}_{k \in K_2}$ is also bounded, there exist an infinite index set $K_3 \subset K_2$ and an accumulation point \hat{d} such that $\lim_{k \in K_3} d_k = \hat{d}$. Taking limit in (3.10) for $k \in K_3$, we obtain

$$-\langle F(\hat{x}), \hat{d} \rangle \leq 0.$$

However, it is easy to see from (2.5) that

$$-\langle F(\hat{x}), \hat{d} \rangle > 0.$$

This yields a contradiction. Consequently, the case (ii) is not possible. The proof is complete. \square

4 Numerical results

In this section, we tested Algorithm 2.1 and compared it with the BFGS method in [13] and the INM method in [10]. We implemented Algorithm 2.1 with the following parameters: $\beta = 0.6$, $\xi = 10^{-8}$ and $\eta = 10^{-4}$. If $\sigma_k \notin [\sigma_{min}, \sigma_{max}]$, we replace it by

$$\sigma_k = \begin{cases} 1 & \text{if } \|F(x_k)\| > 1, \\ \|F(x_k)\|^{-1} & \text{if } 10^{-5} \leq \|F(x_k)\| \leq 1, \\ 10^5 & \text{if } \|F(x_k)\| < 10^{-5}, \end{cases}$$

where $\sigma_{min} = 10^{-10}$ and $\sigma_{max} = 10^{10}$. We stop the iteration if the iteration number exceeds 500 or the inequality

$$\|F(x_k)\| \leq 10^{-4} \quad \text{or} \quad \|F(z_k)\| \leq 10^{-4}$$

is satisfied. The BFGS method in [13] was implemented with the following parameters: $\beta = 0.6$, $\sigma = 10^{-5}$, $h = 10^{-4}$ and $r = 0$. For the INM method in [10], we set $\mu_k = \|F(x_k)\|$, $\rho_k = 0$, $\beta = 0.4$, $\lambda = 0.0001$. The stop criterion is $\|F(x_k)\| \leq 10^{-4}$ or the iteration number exceeds 500. The codes were written in FORTRAN 90 with double precision arithmetic and carried out on a PC (CPU 3.0 GHz, 512M memory) with Windows operation system.

The efficiency of the proposed method was tested on the following two problems with various dimensions and different initial points.

Problem 1 Function F is given by

$$F(x) = Ax + g(x),$$

where $g(x) = (2e^{x_1} - 1, 3e^{x_2} - 1, \dots, 3e^{x_{n-1}} - 1, 2e^{x_n} - 1)^T$ and

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

Problem 2 Function F is given by

$$F_1(x) = 2x_1 + \sin(x_1) - 1$$

$$F_i(x) = -2x_{i-1} + 2x_i + \sin(x_i) - 1.0, \quad i = 2, \dots, n-1$$

$$F_n(x) = 2x_n + \sin(x_n) - 1.$$

We note that Problem 1 is symmetric while Problem 2 is nonsymmetric. The results are listed in Tables 1–2 where $x_1 = (0.1, 0.1, \dots, 0.1)^T$, $x_2 = (1, 1, \dots, 1)^T$, $x_3 = (\frac{1}{n}, \frac{2}{n}, \dots, 1)^T$, $x_4 = (-10, -10, \dots, -10)^T$, $x_5 = (-0.1, -0.1, \dots, -0.1)^T$, $x_6 = (-1, -1, \dots, -1)^T$, $x_7 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^T$, $x_8 = (-0.01, -0.01, \dots, -0.01)^T$, $x_9 = (\frac{n}{n-1}, \frac{n}{n-1}, \dots, \frac{n}{n-1})$, $x_{10} = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$, $x_{11} = (\frac{n}{n+1}, \frac{n}{n+1}, \dots, \frac{n}{n+1})$ and $x_{12} = (\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3})$. In Tables 1–2, we report the problem number along with the initial point number (Pro(initial)), the dimension of each test problem (dim), the number of iterations (iter), the number of function evaluations (fun) and the CPU time in seconds (time). We claim that two method fails, and use the symbol 'F', when some of the following options hold:

- (a) the number of iterations is greater than or equal to 500; or
- (b) the number of backtracking required by the line search along a step is greater than or equal to 20.

We tested each problem 100 times with the same initial point. The CPU time reported in Tables 1–2 is the average value. In the tables, “method 1” and “method 2” represent Algorithm 2.1 and the BFGS method in [13] respectively.

From Tables 1–2, we observe that method 1 performed much better than method 2 and in most cases the INM method performs best. Simultaneously, during the numerical experiments, it is interesting to note that the step $d_k = -F(x_k)$ never appeared when $k > 0$ in method 1. In other words, the condition $d_k^T F_k \leq -\xi \|F_k\|^2$ was always satisfied.

5 Conclusion

In this paper, we propose an algorithm for solving nonlinear monotone equations, which combines the Broyden method and the hyperplane projection method. Under appropriate conditions, we prove that the proposed method is globally convergent. We also report some numerical results to show efficiency of the proposed method.

Table 1 Test results for Problem 1

Method 1				Method 2		INM	
Pro(initial)	Dim	Iter	Time	Iter	Time	Iter	Time
1(1)	(50)	33	0.31250D-02	53	0.78125D-02	26	0.26875D-02
1(2)	(50)	26	0.46875D-02	59	0.10937D-01	21	0.33750D-02
1(3)	(50)	36	0.46875D-02	65	0.93750D-02	25	0.31250D-02
1(4)	(50)	77	0.78125D-02	72	0.12500D-01	101	0.10594D+00
1(5)	(50)	36	0.62500D-02	50	0.93750D-02	35	0.55313D-02
1(6)	(50)	20	0.31250D-02	44	0.46875D-02	121	0.19687D+00
1(7)	(50)	29	0.46875D-02	64	0.10937D-01	24	0.35938D-02
1(8)	(50)	37	0.62500D-02	48	0.78125D-02	92	0.11563D+00
1(9)	(50)	34	0.46875D-02	55	0.10937D-01	32	0.40625D-02
1(10)	(50)	32	0.46875D-02	50	0.93750D-02	36	0.51250D-02
1(11)	(50)	36	0.46875D-02	53	0.78125D-02	30	0.30312D-02
1(12)	(50)	30	0.62500D-02	51	0.93750D-02	23	0.53750D-02
1(1)	(100)	36	0.25000D-01	61	0.40625D-01	23	0.18750D-01
1(2)	(100)	34	0.20313D-01	68	0.45312D-01	26	0.19687D-01
1(3)	(100)	32	0.20313D-01	68	0.42188D-01	23	0.18875D-01
1(4)	(100)	89	0.54688D-01	81	0.54688D-01	121	0.24781D+01
1(5)	(100)	36	0.23438D-01	63	0.39062D-01	28	0.19688D-01
1(6)	(100)	22	0.12500D-01	49	0.31250D-01	116	0.22687D+01
1(7)	(100)	29	0.17188D-01	69	0.42188D-01	22	0.16188D-01
1(8)	(100)	38	0.23438D-01	66	0.42188D-01	99	0.21250D+00
1(9)	(100)	34	0.20313D-01	69	0.45312D-01	27	0.19375D-01
1(10)	(100)	36	0.17188D-01	64	0.40625D-01	27	0.16375D-01
1(11)	(100)	38	0.25000D-01	66	0.40625D-01	36	0.23750D-01
1(12)	(100)	29	0.17188D-01	63	0.39062D-01	22	0.14562D-01
1(1)	(200)	37	0.85938D-01	67	0.16406D+00	28	0.65562D-01
1(2)	(200)	42	0.98437D-01	71	0.19219D+00	23	0.48469D-01
1(3)	(200)	39	0.95312D-01	80	0.20156D+00	24	0.72250D-01
1(4)	(200)	92	0.21563D+00	93	0.23438D+00	113	0.28844D+01
1(5)	(200)	40	0.92188D-01	57	0.14219D+00	47	0.98125D-01
1(6)	(200)	31	0.71875D-01	56	0.14062D+00	91	0.17631D+01
1(7)	(200)	29	0.67187D-01	81	0.20625D+00	18	0.58625D-01
1(8)	(200)	38	0.84375D-01	67	0.16719D+00	85	0.16719D+01
1(9)	(200)	37	0.84375D-01	71	0.18125D+00	43	0.98031D-01
1(10)	(200)	42	0.95312D-01	64	0.16094D+00	24	0.46469D-01
1(11)	(200)	29	0.65625D-01	67	0.17188D+00	17	0.48094D-01
1(12)	(200)	35	0.81250D-01	67	0.17969D+00	22	0.65688D-01

Table 2 Test results for Problem 2

Method 1				Method 2		INM	
Pro(initial)	Dim	Iter	Time	Iter	Time	Iter	Time
2(1)	(20)	105	0.31250D-02	96	0.46875D-02	123	0.25000D-01
2(2)	(20)	95	0.31250D-02	80	0.31250D-02	134	0.25000D-01
2(3)	(20)	82	0.00000D+00	95	0.31250D-02	73	0.15625D-01
2(4)	(20)	120	0.31250D-02	121	0.31250D-02	F	F
2(5)	(20)	94	0.31250D-02	98	0.31250D-02	314	0.46875D-01
2(6)	(20)	91	0.31250D-02	103	0.31250D-02	F	F
2(7)	(20)	75	0.31250D-02	84	0.31250D-02	145	0.25000D-01
2(8)	(20)	96	0.46875D-02	95	0.31250D-02	183	0.31250D-01
2(9)	(20)	65	0.31250D-02	77	0.15625D-02	142	0.21875D-01
2(10)	(20)	74	0.31250D-02	91	0.31250D-02	141	0.18750D-01
2(11)	(20)	73	0.31250D-02	83	0.15625D-02	127	0.21875D-01
2(12)	(20)	87	0.31250D-02	88	0.31250D-02	95	0.18750D-01
2(1)	(50)	227	0.34375D-01	223	0.40625D-01	150	0.24750D-01
2(2)	(50)	177	0.28125D-01	206	0.35937D-01	236	0.23438D+00
2(3)	(50)	175	0.28125D-01	220	0.37500D-01	122	0.15313D-01
2(4)	(50)	F	F	F	F	F	F
2(5)	(50)	170	0.26562D-01	216	0.34375D-01	227	0.40063D-01
2(6)	(50)	197	0.29687D-01	226	0.32813D-01	F	F
2(7)	(50)	311	0.50000D-01	211	0.35937D-01	249	0.43750D-01
2(8)	(50)	170	0.25000D-01	226	0.34375D-01	167	0.19375D-01
2(9)	(50)	164	0.25000D-01	203	0.32813D-01	238	0.22812D+00
2(10)	(50)	209	0.34375D-01	225	0.35937D-01	159	0.18750D-01
2(11)	(50)	273	0.40625D-01	201	0.32813D-01	237	0.35812D-01
2(12)	(50)	210	0.31250D-01	218	0.35937D-01	160	0.27813D-01
2(1)	(100)	F	F	F	F	230	0.10656D+01
2(2)	(100)	F	F	421	0.26719D+00	451	0.30219D+01
2(3)	(100)	460	0.26562D+00	F	F	206	0.18750D+00
2(4)	(100)	174	0.10312D+00	F	F	F	F
2(5)	(100)	378	0.21719D+00	F	F	239	0.14719D+01
2(6)	(100)	483	0.27813D+00	F	F	F	F
2(7)	(100)	F	F	418	0.25938D+00	454	0.27687D+01
2(8)	(100)	365	0.20938D+00	F	F	224	0.10469D+01
2(9)	(100)	F	F	431	0.26875D+00	455	0.36375D+01
2(10)	(100)	370	0.21250D+00	F	F	224	0.10469D+01
2(11)	(100)	F	F	413	0.25469D+00	448	0.29187D+01
2(12)	(100)	379	0.21563D+00	F	F	267	0.11281D+01

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. *Math. Comput.* **19**, 577–593 (1965)
2. Dennis, J.E., More, J.J. Jr.: A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comput.* **28**, 549–560 (1974)
3. Dennis, J.E., More, J.J. Jr.: Quasi-Newton method, motivation and theory. *SIAM Rev.* **19**, 46–89 (1977)
4. Griewank, A.: The global convergence of Broyden-like methods with suitable line search. *J. Aust. Math. Soc. Ser. B* **28**, 75–92 (1986)
5. Gu, G.Z., Li, D.H., Qi, L.Q., Zhou, S.Z.: Descent direction of quasi-Newton methods for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **40**, 1763–1774 (2002)
6. Li, D.H., Fukushima, M.: A globally and superlinear convergent Gauss-Newton based BFGS methods for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **37**, 152–172 (1999)
7. Li, D.H., Fukushima, M.: A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. *Optim. Methods Softw.* **13**, 583–599 (2000)
8. Li, D.H., Cheng, W.Y.: Recent progress in the global convergence of quasi-Newton methods for nonlinear equations. *Hokkaido Math. J.* **36**, 729–743 (2007)
9. Iusem, A.N., Solodov, M.V.: Newton-type methods with generalized distances for constrained optimization. *Optimization* **41**, 257–278 (1997)
10. Solodov, M.V., Svaiter, B.F.: A globally convergent inexact Newton method for systems of monotone equations. In: Fukushima, M., Qi, L. (eds.) *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp. 355–369. Kluwer Academic, Norwell (1998)
11. Martínez, J.M.: Practical quasi-Newton methods for solving nonlinear systems. *J. Comput. Appl. Math.* **124**, 97–121 (2000)
12. Zhao, Y.B., Li, D.: Monotonicity of fixed point and normal mapping associated with variational inequality and its application. *SIAM J. Optim.* **4**, 962–973 (2001)
13. Zhou, W., Li, D.H.: A globally convergent BFGS method for nonlinear monotone equations without any merit functions. *Math. Comput.* **77**, 2231–2240 (2008)